Redundancy Notions for Paramodulation with Non-Monotonic Orderings

Miquel Bofill[†] and Albert Rubio[‡] *

[†]Univ. de Girona,
Dept. IMA,
mbofill@ima.udg.es

[‡]Univ. Politècnica de Catalunya,
Dept. LSI,
rubio@lsi.upc.es

Abstract. Recently, ordered paramodulation and Knuth-Bendix completion have been shown to remain complete when using non-monotonic orderings. However, these results only implied the compatibility with too weak redundancy notions and, in particular, demodulation cannot be applied at all.
In this paper, we present a complete ordered paramodulation calculus compatible with powerful redundancy notions including demodulation, which strictly improves previous results.
Our results can be applied as well to obtain a Knuth-Bendix completion procedure compatible with simplification techniques, which can be used for finding, whenever it exists, a convergent TRS for a given set of equations and a (possibly non-totalizable) reduction ordering.

1 Introduction

Knuth-Bendix-like completion techniques and their extensions to ordered paramodulation for first-order clauses are among the most successful methods for automated deduction with equality [BG98,NR01]. For many years, all known completeness results for Knuth-Bendix completion and ordered paramodulation required the term ordering $\succ$ to be well-founded, monotonic and total (or extendible to a total ordering) on ground terms [HR91,BDH86,BD94,BG94]. In [BGNR99,BGNR03], the monotonicity requirement was dropped and well-foundedness and the subterm property were shown to be sufficient for ensuring refutation completeness of ordered paramodulation (note that any such ordering can be totalized without loosing these two properties). And recently, in [BR02], it was shown that well-foundedness of the ordering suffices for completeness of ordered paramodulation for Horn clauses, i.e., the subterm property can be dropped as well. Apart from its theoretical value, these results have several potential applications in contexts where the usual requirements are too strong. For example, in deduction modulo built-in equational theories E, where E-*compatibility* of the ordering (i.e. $s =_E s' \succ t' =_E t$ implies $s \succ t$) is needed,

* Both authors partially supported by the Spanish CICYT project MAVERISH ref. TIC2001-2476-C03-01. M. Bofill is also supported by the Spanish CICYT project CADVIAL ref. TIC2001-2392-C03-01 and A. Rubio is also supported by the Spanish DURSI group 2001SGR 00254.

finding *E-compatible* orderings fulfilling the required properties is extremely complex or even impossible. For instance, when E contains an idempotency axiom $f(x,x) \simeq x$, no total E-compatible reduction ordering exists: if $s \succ t$, by monotonicity one should have $f(s,s) \succ f(s,t)$, which by E-compatibility implies $s \succ f(s,t)$, and hence non-well-foundedness. Therefore, the techniques for dropping ordering requirements, among other applications, open the door to deduction modulo many more classes of equational theories.

However, ordered strategies are useful for practice only if compatibility with redundancy is shown. Simplification of formulae and elimination of redundant formulae are essential components of automated theorem provers. In fact, in most successful automated theorem provers, simplification is the primary mode of computation, whereas prolific deduction rules are used only sparingly.

Here, we present a calculus for Horn clauses which strictly improves the one in [BGNR03]. On the one hand, regarding the amount of inferences needed to be performed, our inference system is essentially the same as the one in [BGNR03], but, on the other hand, our calculus is compatible with powerful redundancy notions which include simplification by rewriting (i.e. *demodulation*). As a side effect, as in [BGNR03], we can apply our results to obtain a Knuth-Bendix completion procedure, but in this case *compatible with simplification techniques*. This procedure can be used for finding, whenever it exists, a convergent TRS for a given set of equations and a (possibly non-totalizable) reduction ordering.

In our calculus, it is assumed that there is a known reduction ordering $\succ_r$ which is included in the (possibly non-monotonic) ordering $\succ$ w.r.t. which ordered paramodulation is applied. Then we show that redundancy notions w.r.t. the reduction ordering $\succ_r$ can be applied while keeping refutation completeness. Roughly, a formula is redundant if it is a logical consequence of smaller (w.r.t. a well-founded ordering including $\succ_r$) formulae. Then demodulation w.r.t. $\succ_r$ fits in this abstract notion of redundancy since, in particular, a formula which has been simplified by rewriting follows from its smaller w.r.t. $\succ_r$ (due to monotonicity of $\succ_r$) "simplified version" and the (also smaller) equations used to simplify it. As a particular case, if all equations involved in the saturation process[1] turn out to be orientable with $\succ_r$, then demodulation can be fully applied. The following well-known simple challenge example illustrates this situation:

Example 1. Under our inference system,

$$f(f(x)) \simeq f(g(f(x)))$$

is saturated if $f(f(x)) \succ_r f(g(f(x)))$ and the ordering $\succ$ w.r.t. which ordered paramodulation is applied is an extension of $\succ_r$ (which cannot be totalized without loosing monotonicity, since $g(f(x))$ and $f(x)$ should be incomparable in any monotonic extension). The only possible inferences are (i) an ordered paramodulation with $f(f(x)) \simeq f(g(f(x)))$ on itself at the underlined position of $f(\underline{f(x)})$, which has $f(f(g(f(x)))) \simeq f(g(f(f(x))))$ as conclusion and (ii) an

[1] The saturation of a set of clauses S amounts to the closure of S under the inference system *up to redundancy*.

ordered paramodulation with $f(f(x)) \simeq f(g(f(x)))$ on itself at the underlined position of $f(g(\underline{f(x)}))$, which has $f(f(f(x))) \simeq f(g(f(g(f(x)))))$ as conclusion. In either case, the conclusion can be simplified into a tautology by rewriting it with the equation $f(f(x)) \simeq f(g(f(x)))$ (oriented with the reduction ordering). Therefore, the system is saturated. It is worth noting that, without considering demodulation, the closure of $f(f(x)) \simeq f(g(f(x)))$ under ordered paramodulation w.r.t. $\succ$ diverges. □

We must point out that, although demodulation w.r.t. $\succ_r$ is allowed in our inference system, not every clause can be demodulated: during our saturation process, some subterms of clauses become "blocked" for demodulation. Roughly, we block for demodulation those subterms that occur as the small side (w.r.t. $\succ$) of an equation that cannot be handled with $\succ_r$. Later inferences will inherit these blockings. Note that, if paramodulation is applied w.r.t. non-monotonic ordering $\succ$ and we use full demodulation w.r.t. $\succ_r$ we can loose completeness as shown in Example 4 of the Appendix.

The reason for adding these blockings comes from our refutation completeness proof, which is based on the model generation technique of [BG94] and its variant used in [BGNR03]. In the latter, in contrast to the former, the ordering used in the inference system and the ordering used for the induction in the completeness proof do not coincide. In particular, the inferences are applied w.r.t. the non-monotonic ordering $\succ$ and the completeness proof is done by induction on the ordering $\rightarrow_R^+$, where R is the ground rewrite system that defines the model. This is the key for the given completeness proof of ordered paramodulation with non-monotonic orderings. Note that, in this case, since $\succ$ is not monotonic, it may happen that $\rightarrow_R^+$ is not included in $\succ$ although the rules in R are oriented with $\succ$, and in fact it may contradict $\succ$. Due to this lack of compatibility between $\rightarrow_R^+$ and $\succ$, and to the fact that the induction in the completeness proof is done w.r.t. $\rightarrow_R^+$, the redundancy notions should be defined w.r.t. $\rightarrow_R^+$ instead of w.r.t. $\succ$. Unfortunately, $\rightarrow_R^+$ is unknown during the saturation process, and it is not clear how it can be approximated sufficiently. This is why in [BGNR03] it was left open to what extent demodulation could be applied (although some practical redundancy notions like tautology deletion and subsumption were shown to be correct).

Here, our aim is to add for instance demodulation w.r.t. the reduction ordering $\succ_r$, which is included in the non-monotonic ordering $\succ$. The idea for doing that is, roughly, to find a well-founded ordering $\succ_i$, which combines as much as possible $\succ_r$ with $\rightarrow_R^+$, and then prove completeness by induction w.r.t. $\succ_i$. The problem is that, although $\succ_r$ is a well-founded and monotonic ordering and R is a terminating TRS whose rules are included in a well-founded extension $\succ$ of $\succ_r$, the relation $\rightarrow_R \cup \succ_r$ is not well-founded in general, and in fact, $\rightarrow_R^+$ can even contradict $\succ_r$. Take e.g. $f(b) \succ_r f(a)$ and a well-founded extension $\succ$ of $\succ_r$ such that $f(b) \succ f(a) \succ a \succ b$. Then possibly $a \rightarrow b \in R$ (since the rules in R are oriented w.r.t. $\succ$) and hence the infinite sequence $f(a) \rightarrow_R f(b) \succ_r f(a) \cdots$ can be built.

The idea to circumvent this non-well-foundedness problem is to *block* the "bad" right-hand sides, i.e., the ones that can be introduced by a rewriting step with R which is not included in $\succ_r$. Then we can combine $\succ_r$ with $\rightarrow_R$ if the comparisons with $\succ_r$ cannot take into account the blocked terms. For instance, in the previous example, b should be blocked, and then $f(b)$ with b blocked is no longer greater than $f(a)$.

A convenient way to represent terms with blocked positions is by means of superindexed subterms, also called *marked terms*. For example $f(b^x)$, where x is a variable, denotes the term $f(b)$ where the subterm b is blocked. Then, although for performing inferences $f(b^x)$ still corresponds to the term $f(b)$, for the redundancy notions it will be seen as $f(x)$ with a blocked term b in x. In this context $f(b^x)$ cannot be simplified with $f(b) \rightarrow f(a)$, but it could be simplified with $f(x) \rightarrow g(x)$ into $g(b^x)$, if $f(x) \succ_r g(x)$. Marked terms resemble the term closures of [BGLS95]. However, their semantics is fairly different: here the blockings have only effect on the redundancy notions, i.e., the ordered paramodulation inferences are applied in both blocked and unblocked positions.

Detection of terms that have to be marked and introduction of marks in the clauses will be part of our inference system. The terms to be marked come from the equations that cannot be oriented with $\succ_r$ (since we do not know in advance which equations will be part of R). These terms are kept in a set, which is used for introducing marks in the clauses. Hence, if there are many equations that cannot be compared with the reduction ordering then many terms become blocked and we will have almost no redundancy, but the same amount of inferences by paramodulation as in [BGNR03]. On the other hand, if many equations can be handled by the reduction ordering, then few terms will be blocked and we will have as much redundancy as in the paramodulation calculi for total reduction orderings.

The paper is structured as follows. Preliminaries are presented in Section 2. Marked terms and orderings on marked terms are defined in Section 3. In Section 4, our calculus, including the notions of redundancy for inferences, is presented for deduction with sets of equations. It is extended to Horn clauses in Section 5. Saturation derivations including redundancy of clauses are considered in Section 6. In Section 7 some practical notions of redundancy for Horn clauses are presented. In Section 8 the application of our results to obtain a Knuth-Bendix completion procedure for finding convergent TRSs is outlined. Section 9 is devoted to conclusions. Due to lack of space, we have omitted many details and all proofs (see [BR04] for the long version of this paper).

2 Preliminaries

We use the standard definitions of [NR01]. $T(\mathcal{F}, \mathcal{X})$ ($T(\mathcal{F})$) is the set of (ground) terms over a set of symbols $\mathcal{F}$ and a denumerable set of variables $\mathcal{X}$ (over $\mathcal{F}$). By $vars(t)$ we denote the set of variables occurring in a term t. The subterm of t at *position* p is denoted by $t|_p$, the result of replacing $t|_p$ by s in t is denoted by $t[s]_p$, and syntactic equality of terms is denoted by $\equiv$. A *substitution*

is a partial mapping from variables to terms. The application of a substitution σ to a term t is denoted by $t\sigma$. The composition of two substitutions σ_1 and σ_2, denoted by juxtaposition, is defined as the composition of two functions, that is, $t\sigma_1\sigma_2 = (t\sigma_1)\sigma_2$.

An *equation* is a multiset of terms $\{s, t\}$, denoted $s \simeq t$ or, equivalently, $t \simeq s$. A first-order clause is a pair of finite multisets of equations Γ (the *antecedent*) and Δ (the *succedent*), denoted by $\Gamma \to \Delta$. Equations in Γ are called *negative literals* and equations in Δ are called *positive literals*. A Horn clause is a clause with at most one positive literal. The empty clause $\Box$ is a clause where both Γ and Δ are empty.

If $\to$ is a binary relation, then $\leftarrow$ is its inverse[2], $\leftrightarrow$ is its symmetric closure, $\overset{+}{\to}$ is its transitive closure and $\overset{*}{\to}$ is its reflexive-transitive closure. If $s \overset{*}{\to} t$ and there is no t' such that $t \to t'$ then t is called *irreducible* and a *normal form* of s (w.r.t. $\to$). The relation $\to$ is *well-founded* or *terminating* if there exists no infinite sequence $s_1 \to s_2 \to \ldots$ and it is *confluent* or *Church-Rosser* if the relation $\overset{*}{\leftarrow} \circ \overset{*}{\to}$ is contained in $\overset{*}{\to} \circ \overset{*}{\leftarrow}$. A relation $\to$ on terms is *monotonic* if $s \to t$ implies $u[s]_p \to u[t]_p$ for all terms s, t and u and positions p.

A *rewrite rule* is an ordered pair of terms (s, t), written $s \to t$, and a set of rewrite rules R is a *term rewrite system* (TRS). The *rewrite relation* with R on $T(\mathcal{F}, \mathcal{X})$, denoted by $\to_R$, is the smallest monotonic relation such that for all rules $l \to r$ in R, and substitutions σ, $l\sigma \to_R r\sigma$. If $s \to_R t$ then we say that s *rewrites into* t with R. R is called terminating, confluent, etc. if $\to_R$ is.

A (strict partial) *ordering* $\succ$ on $T(\mathcal{F}, \mathcal{X})$ is an irreflexive and transitive binary relation. It is *stable under substitutions* if $s \succ t$ implies $s\sigma \succ t\sigma$ for all substitutions σ. Monotonic orderings that are stable under substitutions are called *rewrite orderings*. A *reduction ordering* is a well-founded rewrite ordering.

A rewrite system terminates if, and only if, its rules are contained in a reduction ordering. In fact, if R is a terminating TRS, then $\to_R^+$ is a reduction ordering. However in restricted cases checking the rules with a well-founded ordering is enough:

Lemma 1. *[BGNR03] Let $\succ$ be a well-founded ordering, and let R be a ground TRS such that, for all $l \to r$ in R, $l \succ r$ and r is irreducible by R at non-topmost positions. Then R is terminating.*

It is said that an ordering fulfills the *subterm property* if $\succ \supseteq \rhd$, where $\rhd$ denotes the strict subterm relation.

The multiset extension of an equivalence relation $\sim$ is denoted by $\sim^{mul}$. The multiset extension of $\succ$ with respect to $\sim$ is defined as the smallest ordering $\succ^{mul}$ on multisets of elements such that

$$M \cup \{s\} \succ^{mul} N \cup \{t_1, \ldots, t_n\} \quad \text{if} \quad M \sim^{mul} N \text{ and } s \succ t_i \text{ for all } i \in 1 \ldots n.$$

If $\succ^{mul}$ is used without explicitly indicating which is the congruence $\sim$ then $\sim$ is assumed to be the syntactic equality relation. If $\succ$ is well-founded on a set S, so is $\succ^{mul}$ on finite multisets over S.

[2] Here, and everywhere, we use the mirror image of a binary relation symbol for its inverse.

There are some orderings that play a central role in our results (in particular, in the redundancy notions). The *subsumption* relation, denoted by $\geq$, is defined as $s \geq t$ if s is an instance of t, i.e., $s \equiv t\sigma$ for some σ. The equivalence relation associated with $\geq$ is denoted by $\doteq$. Note that if $s \doteq t$ then t is a variable renamed version (or a *variant*) of s. The *encompassment* relation, denoted by $\trianglerighteq$, is defined as $s \trianglerighteq t$ if a subterm of s is an instance of t, i.e., $s \trianglerighteq t\sigma$ for some σ; therefore, encompassment is the composition of the subterm and the subsumption relations. If $\succ_r$ is a reduction ordering then $\trianglerighteq \cup \succ_r$ is well-founded.

A *west ordering* [BGNR03] is a well-founded ordering on $T(\mathcal{F})$ that fulfills the subterm property and that is total on $T(\mathcal{F})$ (it is called *west* after well-founded, subterm and total). Every well-founded ordering can be totalized [Wec91], and hence every well-founded ordering satisfying the subterm property can be extended to a west ordering. We also have that every reduction ordering can be extended to a west ordering [BGNR03].

3 Marked terms

Marked terms resemble the closures of [BGLS95]. A *term closure*, following the definition of [BGLS95], is a pair $s \cdot \gamma$ consisting of a term s (the *skeleton*) and a substitution γ from variables to terms. For example, $f(x, g(y)) \cdot \{x \mapsto a, y \mapsto h(z)\}$ is a closure with skeleton $f(x, g(y))$ and substitution $\{x \mapsto a, y \mapsto h(z)\}$. We generalize closures in that the substitution can map a variable not only to a term but also to a closure. For example, $f(x, g(y)) \cdot \{x \mapsto a, y \mapsto h(z) \cdot \{z \mapsto b\}\}$ is a *generalized term closure*, or a *marked term*. Moreover, closures are used in [BGLS95] to express that certain positions are blocked for inferences, whereas in our case they are blocked only for redundancy purposes. In order to ease the reading, sometimes we will denote marked terms by superindexing the "marked" subterms with variables. For example,

$f(x, g(y)) \cdot \{x \mapsto a, y \mapsto h(z) \cdot \{z \mapsto b\}\}$ will be written as $f(a^x, g(h(b^z)^y))$.

Given a substitution γ of the form $\{x_1 \mapsto t_1 \cdot \gamma_1, \ldots, x_n \mapsto t_n \cdot \gamma_n\}$, the domain of γ, denoted by $Dom(\gamma)$, is defined as the set of variables $\{x_1, \ldots, x_n\}$, and the range γ, denoted by $Ran(\gamma)$, is defined as the set of marked terms $\{t_1 \cdot \gamma_1, \ldots, t_n \cdot \gamma_n\}$. The variables occurring in $Dom(\gamma)$ are called *marking variables*. We will restrict marked terms to those which fulfill that marking variables occur at most once in the skeleton. For example $f(x, y) \cdot \{x \mapsto a, y \mapsto a\}$ is a valid marked term, but $f(z, z) \cdot \{z \mapsto a\}$ is not. Therefore, although we generalize closures in one sense, in another sense we restrict them.[3]

Marked terms are extended in the natural way to equations and clauses. An equation between marked terms is called a *marked equation*, and a clause on marked equations is called a *marked clause*. The congruence $\equiv$ on marked terms is defined as the monotonic, reflexive, symmetric and transitive closure including

[3] Both the generalization and the restriction are necessary for our orderings on marked terms being well-founded.

that (i) $t^{y\,x} \equiv t^x$ and (ii) $s[t^x]_p =_m s[t^y]_p$ if y is a variable which does not occur in $s[t^x]_p$. Note that, if $s \cdot \gamma \equiv t \cdot \delta$ then $s \doteq t$, i.e., s is a variant of t.

By $\mathit{Forget}(s \cdot \gamma)$ we denote the term in $T(\mathcal{F}, \mathcal{X})$ obtained from $s \cdot \gamma$ by repeatedly applying all the substitutions. For example, $\mathit{Forget}(f(a^x, g(h(b^z)^y)))$ is $f(a, g(h(b)))$. A marked term $s \cdot \gamma$ is said to be ground if $\mathit{Forget}(s \cdot \gamma)$ is a ground term. This notation is extended to sets of terms, equations and clauses in the usual way.

Given a marked term $s \cdot \gamma$ and a substitution σ (from variables to marked terms), $(s \cdot \gamma)\sigma$ is defined in the natural way but, in order to keep linearity of the marking variables in $(s \cdot \gamma)\sigma$, if in σ we have $\{x \mapsto t \cdot \delta\}$ then we will replace every x by a $t' \cdot \delta'$ obtained by renaming the marking variables of t. If $(s \cdot \gamma)\sigma$ is a ground marked term, then σ is called a *ground substitution* for $s \cdot \gamma$, and $(s \cdot \gamma)\sigma$ is called a *ground instance* of $s \cdot \gamma$. For example,

$$f(x, g(x)^{x'})\{x \mapsto a^y\} = f(a^y, g(a^y)^{x'}) \text{ and } f(x, x)\{x \mapsto a^y\} = f(a^y, a^z).$$

Note that, in the latter case, a marking variable has been renamed to keep linearity of marking variables in the skeleton.

The usual concepts of (and notation for) position, subterm and subterm replacement can be easily extended for marked terms. For example,

$$f(x, h(a)^y)|_2 = h(a)^y,\ f(x, h(a)^y)|_{2\cdot 1} = a \text{ and } f(x, h(a)^y)[b^z]_{2\cdot 1} = f(x, h(b^z)^y).$$

We write $u \trianglerighteq t \cdot \delta$ if $t \cdot \delta$ occurs in u, i.e., $u|_p \equiv t \cdot \delta$ for some position p, and we write $u \rhd t \cdot \delta$ if $p \neq \lambda$. From these definitions, the notion of rewriting for terms can be extended to marked terms in the natural way.

Recall that, given two terms s and s' in $T(\mathcal{F}, \mathcal{X})$, we write $s \doteq s'$ if s' is a variant (that is, a variable renamed version) of s. This extends to marked terms by writing $s \cdot \gamma \doteq s' \cdot \gamma'$ if there are some substitutions σ and σ' such that $(s \cdot \gamma)\sigma \equiv s' \cdot \gamma'$ and $s \cdot \gamma \equiv (s' \cdot \gamma')\sigma'$. For example, if x, y, z and z' are variables and b is not a variable, then $f(g(z)^x) \doteq f(g(z')^y)$, whereas $f(g(b)^x) \not\doteq f(g(b^z)^y)$.

The concept of unifier between terms is extended to marked terms: a unifier of two marked terms $s \cdot \gamma$ and $t \cdot \delta$ is a substitution σ such that $(s \cdot \gamma)\sigma \equiv (t \cdot \delta)\sigma$. There exists a unique (up to renaming of variables) most general unifier between marked terms, which can be obtained by straightforwardly adapting the known unification algorithms for terms.

3.1 Orderings on marked terms

In this section we will define the two main orderings on martked terms, $\succ_{red}$ which will be used in the redundancy notions and $\succ_i$ which will be used as induction ordering in the completeness proof. For the reasons explained in the introduction, it is important that $\succ_{red} \subseteq \succ_i$.

Let $\succ_r$ be a reduction ordering, let $\succ$ be a west ordering including $\succ_r$ and let R be a terminating ground TRS.

First we define the marking ordering $\succ_m$. It only applies to terms that just differ on the markings and it is similar to the subsumption ordering for terms, in the sense that a term is smaller if it has higher marks.

Definition 1. *Let $s\cdot\gamma$ and $t\cdot\delta$ be two marked terms. Then $s\cdot\gamma \succ_m t\cdot\delta$ if and only if $Forget(s\cdot\gamma) \equiv Forget(t\cdot\gamma)$ and (i) $s \trianglerighteq t$ or (ii) $s \doteq t$ and $Ran(\gamma) \succ_m^{mul} Ran(\delta)$.*

The ordering $\succ_{red}$ will be used in the redundancy notions. Due to this all its ingredients are known while performing paramodulation steps.

Definition 2. *Let $s \cdot \gamma$ and $t \cdot \delta$ be two marked terms. Then $s \cdot \gamma \succ_{red} t \cdot \delta$ if and only if (i) $s \succ_r \cup \rhd t$ or (ii) $s \doteq t$ and $Forget(s \cdot \gamma) \equiv Forget(t \cdot \gamma)$ and $Ran(\gamma) \succ_m^{mul} Ran(\delta)$.*

The ordering $\succ_i$ will be used in the induction proof. For this reason, it extends $\succ_{red}$ with an additional case which uses as ingredient a rewrite system R that will be known only in the proof.

Definition 3. *Let $s \cdot \gamma$ and $t \cdot \delta$ be two marked terms. Then $s \cdot \gamma \succ_i t \cdot \delta$ if and only if (i) $s \succ_r \cup \rhd t$ or (ii) $Forget(s\cdot\gamma) \equiv Forget(t\cdot\gamma)$ and $Ran(\gamma) \succ_m^{mul} Ran(\delta)$ or (iii) $s \doteq t$ and $Forget(Ran(\gamma)) \rightarrow_R^+ Forget(Ran(\delta))$.*

In what follows, we will consider $\succ_i$ and $\succ_{red}$ as the transitive closures of the orderings defined above.

Lemma 2. *$\succ_i$ is well-founded.*

Example 2. Here there are several examples of applications of $\succ_i$.

1. If $f(x) \succ_r g(x)$ for all x, then $f(a) \succ_i f(a^x) \succ_i g(a^x) \succ_i a^x$ by case (i) of the definition of $\succ_i$.
2. We have $h(g(f(a^x, a))^y) \succ_i h(g(f(a^x, a^z))^y) \succ_i h(g(f(a, a)^x)^y)$ by case (ii) of the definition of $\succ_i$.

4 Paramodulation with equations

In this section we present an ordered paramodulation calculus for sets of equations. The calculus will work on both a set of marked equations E and a set of terms T, which represents the set of terms that should be marked.

In what follows, we assume that the non marking variables of all equations are disjoint (if necessary, the variables can be renamed). On the other hand, to ease the reading, the marked terms of an equation are considered to have the same substitution (if necessary, the substitution can be extended and the marking variables of one of the two skeletons can be renamed).

Definition 4. *We assume that we have a set E of equations on marked terms and a set T of terms which satisfy the following condition: for every marked subterm $(s \cdot \gamma)^x$ occurring in E, we have that $Forget(s \cdot \gamma)$ is an instance of a term in T. Our inference system $\mathcal{I}$ consists of the following two inference rules and a mark detection rule:*

Paramodulation*:*

$$\frac{l \cdot \delta \simeq r \cdot \delta \qquad s \cdot \gamma \simeq t \cdot \gamma}{(s \cdot \gamma[r \cdot \delta]_p \simeq t \cdot \gamma)\sigma} \quad \textit{if}$$

1. $\sigma = mgu(l \cdot \delta, s \cdot \gamma|_p)$, *the most general unifier of* $l \cdot \delta$ *and* $s \cdot \gamma|_p$,
2. $Forget(s \cdot \gamma|_p)$ *is not a variable,*
3. $l \succ_r r$ *or* $r \cdot \delta$ *is marked on top, and*
4. *for some ground substitution* θ, *we have* $Forget((l \cdot \delta)\sigma\theta) \succ Forget((r \cdot \delta)\sigma\theta)$ *and, if* $p = \lambda$, *then we also have* $Forget((s \cdot \gamma)\sigma\theta) \succ Forget((t \cdot \gamma)\sigma\theta)$.

Marking*:*

$$\frac{s \cdot \gamma[u \cdot \delta]_p \simeq t \cdot \gamma}{(s \cdot \gamma[(u \cdot \delta)^x]_p \simeq t \cdot \gamma)\sigma} \qquad \textit{if}$$

1. $\sigma = mgu(Forget(u \cdot \delta), u')$ *for some term* u' *in* T,
2. u *is not a variable, and*
3. x *is a fresh variable.*

Mark detection*:*

$$T \Longrightarrow T \cup \{Forget(r \cdot \delta)\} \qquad \textit{if}$$

1. $l \cdot \delta \simeq r \cdot \delta \in E$,
2. $l \not\succ_r r$ *and* $r \cdot \delta$ *is not marked on top, and*
3. *for some ground substitution* θ, *we have that* $Forget((l \cdot \delta)\theta) \succ Forget((r \cdot \delta)\theta)$.

Some examples of applications of these rules can be found in the Example 4 of the Appendix.

4.1 Redundancy notions: a static view

First we will define some abstract redundancy notions for inferences like the ones in [BG94] and [NR01]. Some practical redundancy notions fitting in these abstract notions will be discussed afterwards, in Section 7.

Given a set of terms T and a set of equations E, by $gnd_T(E)$ we denote the set of all ground instances of equations in E such that all marks correspond to terms in T. Let $>$ be a well-founded ordering on equations. By $E^{<e}$ we denote the set of all d in E such that $e > d$.

Now we define redundancy of inferences and mark detection rules.

Definition 5. *Let* π *be an inference by* $\mathcal{I}$ *with premises* $C_1, \ldots C_n$ *and conclusion* D. *Then a* ground instance $\pi\theta$ of the inference π *is an inference by* $\mathcal{I}$ *with premises* $C_1\theta, \ldots C_n\theta$ *and conclusion* $D\theta$ *for some ground* θ.

Let G *be a set of ground marked equations. A ground inference by* $\mathcal{I}$, *with premises* $C_1, \ldots C_n$ *and conclusion* D *is* redundant *in* G *if* $Forget(G^{\prec^{mul}_{red} C_n}) \cup Forget(G^{\preceq^{mul}_{red} D}) \models Forget(D)$.

Let E *a set of marked equations. An inference by* $\mathcal{I}$, *with premises* $C_1, \ldots C_n$ *and conclusion* D *is* redundant *in* E *if, for all sets of terms* M *and ground substitutions* θ *with marks in* M, $\pi\theta$ *is redundant in* $gnd_M(E)$.

Let T *be a set of terms. A mark detection rule* $T \Longrightarrow T \cup \{t\}$ *is redundant in* T *if* t *is an instance of a term in* T.

Note that the set of terms M is universally quantified in the definition of redundancy of inferences. This is due to the fact that the set of terms T to be marked will change during the computation, and it is crucial to have a notion redundancy which is preserved along the computation. Moreover, since in general the conclusion of an inference does not need to be smaller w.r.t. $\succ_{red}^{mul}$ than the rightmost premise, the set $G^{\preceq_{red}^{mul} D}$ may not be included in $G^{\prec_{red}^{mul} C_n}$.

Redundant inferences and mark detection rules are unnecessary, and therefore we are interested in computing the closure of the set of equations with respect to $\mathcal{I}$ up to redundancy.

Definition 6. *Let E be a set of equations and T be a set of terms. The pair $\langle E, T\rangle$ is* saturated *with respect to $\mathcal{I}$ if every inference by $\mathcal{I}$ is redundant in E and every mark detection rule is redundant in T.*

4.2 Model generation

Given a set of equations E on marked terms and a set of terms T, we show how to build an equational model of E defined by means of a set of rules coming from ground instances of E and T. Since the model is defined inductively, we need to treat E and T in a uniform way. To this end, a new constant symbol $\bot$ is introduced and the west ordering is extended with $\bot$ as minimal element.

Now, let T_E be the set of equations $t \simeq \bot$ for all $t \in T$, and let T_G be the set of mark free ground instances of equations in T_E. The model is defined inductively on the equations in T_G and $gnd_T(E)$ with respect to $\succ_{mod}$, a total well-founded ordering including: $(s \simeq t) \succ_{mod} (u \simeq v)$ if

1. $Forget(s \simeq t) \succ^{mul} Forget(u \simeq v)$.
2. $Forget(s \simeq t) \equiv^{mul} Forget(u \simeq v)$ and $(s \simeq t) \succ_m^{mul} (u \simeq v)$, which means that $u \simeq v$ has marks in higher positions than $s \simeq t$.

Now we define a rewrite system R_E from $gnd_T(E)$, which requires to define at the same time a rewrite system R_T from T_G.

Definition 7. *The rewrite system R_{ET} is the union of R_E and R_T. Let e be a ground (marked) equation in $gnd_T(E) \cup T_G$.*

1. *If e is an instance $(s \cdot \gamma)\theta \simeq (t \cdot \gamma)\theta$ in $gnd_T(E)$ of some equation $s \cdot \gamma \simeq t \cdot \gamma$ in E then it generates the rule $(s \cdot \gamma)\theta \to (t \cdot \gamma)\theta$ in R_E if*
   (a) $s \succ_r t$ or $t \cdot \gamma$ is marked on top,
   (b) $Forget((s \cdot \gamma)\theta) \succ Forget((t \cdot \gamma)\theta)$,
   (c) $(s \cdot \gamma)\theta$ is irreducible by $R_E^e \cup R_T^e$, and
   (d) $(t \cdot \gamma)\theta$ is irreducible by $R_E^e \cup R_T^e$ at non-topmost positions.
2. *If e is an instance $t\theta \simeq \bot$ in T_G of an equation $t \simeq \bot$ in T_E then it generates a rule $u \cdot \delta \to (u \cdot \delta)^x$ if $u \cdot \delta$ is the normal form of $t\theta$ w.r.t. R_T^e and $u \cdot \delta$ is irreducible by R_E^e,*

where R_E^e and R_T^e denote the set of rules in R_E and R_T respectively, generated by equations d such that $e \succ_{mod} d$.

The following lemma about irreducibility of left and right-hand sides of R_{ET} holds easily by construction.

Lemma 3. *For all rules $l \cdot \delta \to r \cdot \delta$ in R_{ET} we have that*

1. *$r \cdot \delta$ is irreducible by R_{ET} at non-topmost positions, and*
2. *$l \cdot \delta$ is irreducible by $R_{ET} \setminus \{l \cdot \delta \to r \cdot \delta\}$.*

From the previous lemma, convergence of R_{ET} follows.

Lemma 4. *R_{ET} is convergent.*

By construction and the fact all marks are instances of terms in T, we obtain the following.

Lemma 5. *Let u be a marked subterm occurring in R_E. Then u is of the form $(s \cdot \gamma)^x$ if, and only if, $Forget(u)$ is an instance of a term in T.*

Then, from lemmas 3 and 5, we obtain the same irreducibility result for $Forget(R_E)$ as for R_{ET}, which easily implies the convergence of $Forget(R_E)$.

Lemma 6. *$Forget(R_E)$ is convergent.*

Finally, we show that $Forget(R_E)^*$ is a model of $Forget(E)$.

Theorem 1. *If a pair $\langle E, T\rangle$ is saturated with respect to $\mathcal{I}$, then $Forget(R_E)^* \models Forget(E)$.*

Once we have that $Forget(R_E)$ is convergent, the completeness proof is an straightforward adaptation of the one in [BG94,NR01], but where the induction is done w.r.t. $\succ_i^{mul}$, taking $R = Forget(R_E)$ (see Section 3.1 for the definition of $\succ_i$). A contradiction is derived from the existence of a minimal w.r.t. $\succ_i^{mul}$ ground instance $e\theta$ in $gnd_T(E)$ of an equation e in E, such that $R^* \not\models Forget(e\theta)$. As usual, we have that $e\theta$ has not generated a rule in R_E. If this is because $e\theta$ is reducible by R_E, then we have an inference by paramodulation and, if $e\theta$ is reducible by R_T, then we have an inference by marking and, in both cases, a contradiction is obtained.

5 Horn clauses

In this section we generalize the results of the previous section to Horn clauses. In the inference system it is assumed that in each clause with a non-empty antecedent one of these negative equations, the one that is written underlined, has been *selected* (see, e.g., [BG98]). We assume that we have a set of Horn clauses S and a set of terms T.

The inference system $\mathcal{H}$ for Horn clauses consists of five inference rules, namely, *paramodulation right*, *paramodulation left*, *equality resolution*, *marking right* and *marking left*, and a *mark detection* rule. Paramodulation and marking

right, and the mark detection rule, coincide with the ones of the equational case, since they are applied only to positive unit clauses (i.e., equations). Marking left is the same as marking right, but applied on the selected negative equation of the clause. Paramodulation left is similar to paramodulation right, but applied on the selected negative equation. And equality resolution is defined as usual but using unification between marked terms. Due to lack of room, we only give the definition of these two latter inference rules (see Example 4 of the Appendix for some examples of applications of these rules).

Paramodulation left:

$$\frac{\rightarrow l \cdot \delta \simeq r \cdot \delta \qquad \Gamma, \underline{s \cdot \gamma \simeq t \cdot \gamma} \rightarrow \Delta}{(\Gamma, s \cdot \gamma[r \cdot \delta]_p \simeq t \cdot \gamma \rightarrow \Delta)\sigma} \qquad \text{if}$$

1. $\sigma = mgu(l \cdot \delta, s \cdot \gamma|_p)$,
2. $Forget(s \cdot \gamma|_p)$ is not a variable,
3. $l \succ_r r$ or $r \cdot \delta$ is marked on top, and
4. for some ground substitution θ, we have $Forget((l \cdot \delta)\sigma\theta) \succ Forget((r \cdot \delta)\sigma\theta)$.

Equality resolution:

$$\frac{\Gamma, \underline{s \cdot \gamma \simeq t \cdot \gamma} \rightarrow \Delta}{\langle(\Gamma \rightarrow \Delta)\sigma} \qquad \text{if } \sigma = mgu(s \cdot \gamma, t \cdot \gamma).$$

We use standard multiset extensions to lift the orderings from marked equations to marked clauses. Then the notions of redundancy and saturation are directly obtained from the ones given in Section 4.1 by using Horn clauses instead of equations and the inference system $\mathcal{H}$ instead of $\mathcal{I}$.

Using lemmas 5 and 6 and Theorem 1 applied to the rewrite model obtained from the positive unit clauses, the following completeness theorem is obtained in the usual way.

Theorem 2. *(Completeness) Let S be a set of Horn clauses and T a set of terms such that the pair $\langle S, T\rangle$ is saturated with respect to $\mathcal{H}$. Then $\Box \in S$ if, and only if, $Forget(S)$ is unsatisfiable.*

6 Derivations and redundancy of clauses

In the previous section, we showed that an unsatisfiable saturated set of (Horn) clauses contains the empty clause. Saturatedness was defined in terms of redundancy of inferences. Here, we address the problem of how to compute such saturated sets. Usually, a saturation procedure is modeled by means of a *derivation*, a sequence of sets of clauses where each set can be obtained from the previous either by adding a logical consequence or by removing some redundant clause. In our case, a ground clause C is defined to be *redundant* in a set of ground marked clauses G if $Forget(G^{\prec^{mul}_{red} C}) \models Forget(C)$. Then a clause C is *redundant* in a set of marked clauses S if, for all sets of terms M and ground

substitutions θ with marks in M, $C\theta$ is redundant in $gnd_M(S)$. Since we deal with a set of clauses S and a set of terms T, the notion of derivation is adapted as follows: a *derivation* is a (possibly infinite) sequence of pairs of sets of clauses and terms $\langle S_0, T_0\rangle, \langle S_1, T_1\rangle, \ldots$ where for each $\langle S_{i+1}, T_{i+1}\rangle$ we have either

1. $T_{i+1} = T_i$ and
   (a) $S_i \cup \{C\}$, for some C such that $\mathit{Forget}(S_i) \models \mathit{Forget}(C)$, or
   (b) $S_i \setminus \{C\}$, for some C which is redundant in S_i, or
2. $S_{i+1} = S_i$ and
   (a) $T_{i+1} = T_i \cup \{t\}$ for some term t, or
   (b) $T_{i+1} = T_i \setminus \{t\}$ where t is an instance of some term in $T_i \setminus \{t\}$.

It is well-known that simplification by rewriting fits into the notion of derivation [NR01]. As shown in the following section we can add the simplified clause and then remove the original clause, which has become redundant.

Clauses belonging, from some i on, to all S_k with $k > i$, are called *persistent*. The set of persistent clauses S_∞ is formally defined as $S_\infty = \cup_i \cap_{k>i} S_k$. Analogously, the set of persistent terms is defined as $T_\infty = \cup_i \cap_{k>i} T_k$. In order to obtain a saturated set in the limit of a derivation, some notion of *fairness* of the derivation is required. We will use the following definition: a derivation $\langle S_0, T_0\rangle, \langle S_1, T_1\rangle, \ldots$ is *fair* with respect to an inference system $\mathcal{I}nf$ if (i) every inference using clauses in S_∞ and terms T_∞ is redundant in some S_j and (ii) every mark detection rule using clauses in S_∞ is redundant in some T_j.

Lemma 7. *If $\langle S_0, T_0\rangle, \langle S_1, T_1\rangle, \ldots$ is a fair derivation with respect to an inference system $\mathcal{I}nf$, then $\langle S_\infty, T_\infty\rangle$ is saturated with respect to $\mathcal{I}nf$.*

Then, from Lemma 7 and Theorem 2, it follows the refutation completeness of any theorem proving procedure which computes a fair derivation w.r.t., for example, the inference system $\mathcal{H}$. Note that, in order to ensure fairness in practice, in the inference systems presented in this paper it suffices that no inference (or mark detection rule) with persistent premises is postponed infinitely many times, since adding (a simplified version of) the conclusion always makes the inference or mark detection rule redundant.

As usual, useful methods like *subsumption* of clauses can be covered for instance, by more general notions of redundancy for clauses like *non-strict redundancy*, together with the notion of *ground fairness* [NR01] (see also [BG94] for an alternative way of capturing subsumption).

7 Practical notions of redundancy

In this section we provide sufficient conditions for proving redundancy of inferences, which includes simplification by rewriting for those rules included in the reduction ordering $\succ_r$.

Let S be a set of Horn clauses. By S_r we denote the set of rules $l \cdot \delta \to r \cdot \delta$ such that the clause $\to l \cdot \delta \simeq r \cdot \delta$ is in S and $l \succ_r r$.

An inference with premises $C_1, \ldots, C_n$ and conclusion D can be show redundant in S in two steps:

1. First, D is simplified by rewriting it into D' using marked rules $l \cdot \delta \to r \cdot \delta$ in S_r. In order to ensure that we are only using smaller (w.r.t. $\succ_{red}^{mul}$) instances of equations in S, we have the following sufficient conditions. If D is of the form $\Gamma \to s \simeq t$, then (i) we can apply any rewriting step on Γ, but only at positions of the skeletons; (ii) we can apply any rewriting step at a non-top position of the skeleton of s (resp. t) obtaining s' (resp. t'), and after that we can apply any rewriting step at any position of the skeleton of s' (resp. t'); (iii) we can apply a rewriting step at top position of s (resp. t) obtaining s' (resp. t') if the instance of the rule applied is smaller (w.r.t. $\succ_{red}^{mul}$) than C_n or smaller than or equal to D, and after that we can apply any rewriting step at any position of the skeleton of s' (resp. t'). Note that in all these cases we can show that for any ground instance of the inference, the ground instances of the applied rules are either smaller (w.r.t. $\succ_{red}^{mul}$) than the instance of C_n or smaller than or equal to the instance of D.
2. Second, we check whether either $D' \in S$ or $D_1, \ldots, D_n \models D'$ for some instances of clauses in S s.t. $D' \trianglerighteq_{sk}^{mul} \cup \succ_m^{mul} D_i$ for all i in $1 \ldots n$, where $u \trianglerighteq_{sk} t \cdot \delta$ if $t \cdot \delta$ occurs in u at some position of its skeleton. This includes, in particular, the case where D' is a tautology. If none of both situations applies then we can add D' to S to make D redundant.

All previous results can be adapted in an easy way for showing redundancy of a clause C by using only clauses smaller (w.r.t. $\succ_{red}^{mul}$) than C.

Example 3. In example 1 of the introduction we showed how, by an ordered paramodulation with $\underline{f(f(x))} \simeq f(g(f(x)))$ on $f(f(y)) \simeq f(g(\underline{f(y)}))$, we got the equation $f(f(f(x))) \simeq \overline{f(g(f}(g(f(x)))))$, i.e. $f(f(f(z))) \simeq f(\overline{g(f(}g(f(z)))))$. In this case, $f(f(f(z)))$ could be rewritten in two steps with $f(f(x)) \to f(g(f(x)))$ into $f(g(f(g(f(z)))))$, and hence the conclusion of the inference could be simplified into a tautology. Observe that the first step is applied at a non-top position, obtaining $f(f(g(f(z))))$, and then the second step can be applied on top obtaining $f(g(f(g(f(z)))))$.

8 Knuth-Bendix completion

In this section, E denotes a set of equations and $\succ_r$ denotes a (possibly non-totalizable) reduction ordering on $T(\mathcal{F}, \mathcal{X})$. Then R is a *convergent TRS for E and $\succ_r$* if it is convergent, logically equivalent to E, and $l \succ_r r$ for all its rules $l \to r$. The problem we deal with is finding a convergent TRS for the given E and $\succ_r$ whenever it exists, and find it in finite time if it is finite.

It is not difficult to devise a procedure of theoretical nature for doing this by enumerating all equational consequences of E. However, finding practically useful procedures remained as an open problem for a long time. The first procedure for the general problem, not relying on the enumeration of all equational consequences, was presented in [BGNR99,BGNR03]. Here we improve the method presented in [BGNR03] by making it compatible with powerful redundancy notions, like simplification by rewriting. The proof is an straightforward adaptation

of the one in [BGNR03] since, in the limit, an interreduced convergent TRS can be generated and, hence, the same techniques can be applied.

In the following, R denotes the canonical TRS for E and $\succ_r$ (we assume that R exists), and $\succ$ denotes a west ordering extending $\succ_r$. We will use the same inference system as in Definition 4, but where the *Paramodulation* inference is relaxed by changing condition 4 as follows:

4. for some ground substitution θ, $\mathit{Forget}((l \cdot \delta)\sigma\theta) \succ \mathit{Forget}((r \cdot \delta)\sigma\theta)$.

With this change, now the ordered paramodulation rule is also applied on top of the small sides. This is necessary in order to obtain an interreduced ground TRS for the model. We denote the resulting inference system by $\mathcal{K}$.

By R_E, R_T and R_{ET} we denote the $TRSs$ generated from a set of marked equations E and a set of terms T as in Definition 7, but where in case 1d total irreducibility is required instead of just irreducibility at non-topmost positions. Then it follows that $\mathit{Forget}(R_E)$ is interreduced. Moreover, since an equation $s \simeq t$ can be identified with a clause $\rightarrow s \simeq t$, the dynamic redundancy notions and lemmas for clauses of section 6 hold for equations as well. Then, since $\mathit{Forget}(R_E)^* \models \mathit{Forget}(E)$ if $\langle E, T\rangle$ is saturated w.r.t. $\mathcal{K}$, by the same arguments as in [BGNR03] we get the following result:

Theorem 3. *Let $\langle E_0, T_0\rangle, \langle E_1, T_1\rangle, \ldots$ be a fair derivation with respect to $\mathcal{K}$, where E_0 is the given set of equations E. Then $\mathit{Forget}(E_\infty) \supseteq R$.*

Therefore, if R is finite, it will be contained in $\mathit{Forget}(E_j)$ for some E_j of the derivation.

9 Conclusions

We have proposed a refutation complete ordered paramodulation inference system for Horn clauses which works with non-monotonic orderings and is compatible with powerful redundancy notions. Our inference system work with a pair of orderings: a west ordering $\succ$ which is used in the inferences and a reduction ordering $\succ_r$ included in $\succ$ which is used in the redundancy notions. Our method is based on adding some marks that block terms for redundancy (but not for inferences). We believe that these results extend to general clauses, but maybe some more mark detection rules need to be added.

Using these results in the same way as in [BGNR03] we obtain a Knuth-Bendix completion procedure, *compatible with simplification by rewriting*, that finds a convergent TRS for a given set of equations E and a (possibly non-totalizable) reduction ordering $\succ_r$ whenever it exists.

As a final remark, we want to note that a way to obtain, in practice, the orderings $\succ$ and $\succ_r$ is to define the reduction ordering $\succ_r$ with the *monotonic semantic path orderings* [BFR00], since then it is directly included, by definition, in a *semantic path ordering* [KL80], which can be used as the west ordering $\succ$.

References

[BD94] L. Bachmair and N. Dershowitz. Equational inference, canonical proofs, and proof orderings. *J. of the Association for Computing Machinery*, 41(2):236–276, 1994.

[BDH86] L. Bachmair, N. Dershowitz, and J. Hsiang. Orderings for equational proofs. In *First IEEE Symposium on Logic in Computer Science (LICS)*, pp. 346–357, 1986. IEEE Computer Society Press.

[BFR00] C. Borralleras, M. Ferreira, and A. Rubio. Complete monotonic semantic path orderings. *Proc. of the 17th International Conference on Automated Deduction*, LNAI 1831, pp. 346–364, 2000. Springer-Verlag.

[BG94] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.

[BG98] L. Bachmair and H. Ganzinger. Equational reasoning in saturation-based theorem proving. In W. Bibel and P. Schmitt, editors, *Automated Deduction: A Basis for Applications*. Kluwer, 1998.

[BGLS95] L. Bachmair, H. Ganzinger, Chr. Lynch, and W. Snyder. Basic paramodulation. *Information and Computation*, 121(2):172–192, 1995.

[BGNR99] M. Bofill, G. Godoy, R. Nieuwenhuis, and A. Rubio. Paramodulation with non-monotonic orderings. In *14th IEEE Symposium on Logic in Computer Science*, pp. 225–233, Trento, Italy, July 2–5, 1999.

[BGNR03] M. Bofill, G. Godoy, R. Nieuwenhuis, and A. Rubio. Paramodulation and Knuth-Bendix Completion with Nontotal and Nonmonotonic Orderings. *Journal of Automated Reasoning*, 30(1):99–120, 2003.

[BR02] M. Bofill and A. Rubio. Well-foundedness is sufficient for completeness of ordered paramodulation. In *Proc. of the 18th International Conference on Automated Deduction*, LNAI 2392, pp.456–470, 2002. Springer-Verlag.

[BR04] M. Bofill and A. Rubio. Redundancy notions for paramodulation with non-monotonic orderings. Available at `www.lsi.upc.es/~albert/papers.html`, 2004.

[HR91] J. Hsiang and M Rusinowitch. Proving refutational completeness of theorem proving strategies: the transfinite semantic tree method. *Journal of the ACM*, 38(3):559–587, 1991.

[KL80] S. Kamin and J.-J. Levy. Two generalizations of the recursive path ordering. Unpublished note, Dept. of Computer Science, Univ. of Illinois, 1980.

[NR01] Robert Nieuwenhuis and Albert Rubio. Paramodulation-based theorem proving. In J.A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume 1, chapter 7, pages 372–444. Elsevier Science Publishers and MIT Press, 2001.

[Wec91] W. Wechler. *Universal Algebra for Computer Scientists*, volume 25 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Berlin, 1991.

Appendix

In the following example we show how in our setting, where ordered paramodulation is applied w.r.t. a (possibly non-monotonic) west ordering $\succ$ extending a reduction ordering $\succ_r$, demodulation w.r.t. $\succ_r$ can cause incompleteness if no blockings are introduced at all. This thus shows the need for some control or, in other words, that our blockings are not merely a technicality to prove completeness.

Example 4. Let $\succ_r$ be a reduction ordering such that

$$f(x,x) \succ_r f(a,b) \text{ and}$$
$$f(x,x) \succ_r f(b,a).$$

Then necessarily

$$a \not\succ_r b \text{ and}$$
$$b \not\succ_r a.$$

Let S denote the following inconsistent set of Horn clauses:

$$\begin{array}{ll} 1) & \rightarrow a \simeq b \\ 2) & \rightarrow f(x,x) \simeq f(a,b) \\ 3) & \rightarrow f(x,x) \simeq f(b,a) \\ 4) & \rightarrow f(x,x) \simeq f(b,b) \\ 5) & \rightarrow g(f(a,b)) \simeq g(g(f(b,a))) \\ 6) & x \simeq g(x) \rightarrow \end{array}$$

First of all, we will show how the empty clause can be derived under our inference system (which introduces the convenient blockings). Let $\succ$ be a west ordering which is an extension of $\succ_r$ and such that includes

$$g(g(f(a,b))) \succ g(f(a,b)) \succ g(g(f(b,a))) \succ g(f(b,a))$$
$$\succ f(a,a) \succ f(b,b) \succ f(a,b) \succ f(b,a) \succ a \succ b.$$

Initially, the set T of terms that should be marked is empty. Then, since $a \succ b$ and $a \not\succ_r b$, the *Mark detection* rule gives us $T = \{b\}$. After this, an inference by *Marking right* into 1 gives us

$$7) \quad \rightarrow a \simeq b^{x_0}$$

Now, an inference by *Paramodulation right* with 7 into 5 is possible (recall that an inference by paramodulation right with 1 into 5 is not possible since $a \simeq b$ does not fulfill $a \succ_r b$ and b is not marked on top), giving

$$8) \quad \rightarrow g(f(b^{x_1}, b)) \simeq g(g(f(b,a)))$$

A new inference by *Paramodulation right* with 7 into 8 gives us

$$9) \quad \rightarrow g(f(b^{x_1}, b)) \simeq g(g(f(b, b^{x_4})))$$

And, with two more inferences by *Marking right*, we get

$$10) \quad \rightarrow g(f(b^{x_1}, b^{x_2})) \simeq g(g(f(b, b^{x_4})))$$
$$11) \quad \rightarrow g(f(b^{x_1}, b^{x_2})) \simeq g(g(f(b^{x_3}, b^{x_4})))$$

It is worth noting that clause 1 follows from the smaller clause 7, clauses 9 and 10 follow from the smaller clause 11, and clause 8 follows from the smaller clauses 7 and 11. Hence, using redundancy of clauses 1, 8, 9 and 10 could be deleted. Redundancy of clauses is considered in Section 6.

Now, recall that $\succ$ fulfills the subterm property. Therefore we have that $\mathit{Forget}(g(g(f(b^{x_3}, b^{x_4})))) \succ \mathit{Forget}(g(f(b^{x_1}, b^{x_2})))$, and hence an inference by *Paramodulation left* with 11 into 6 gives us

$$12) \quad g(f(b^{x_1}, b^{x_2})) \simeq g(f(b^{x_1}, b^{x_2})) \rightarrow$$

And finally, the empty clause is obtained with an inference by *Equality resolution*.

Now let us show that, if the blockings were not introduced (in other words, if inferences by *Paramodulation right* with 1 instead of 7 were allowed), then refutation completeness would be lost. If an inference by *Paramodulation right* with 1 into 5 was possible, then it would give either

$$\rightarrow g(f(b, b)) \simeq g(g(f(b, a)))$$

or

$$\rightarrow g(f(a, b)) \simeq g(g(f(b, b))),$$

which could be simplified back into 5 respectively with 2 and 3, which are included in the reduction ordering. Moreover, inferences by *Paramodulation right* with 1 into 2 and 3 give us 4. And inferences by *Paramodulation right* between 2, 3 and 4 give us already existing clauses or clauses like

$$\rightarrow f(a, b) \simeq f(b, a),$$

which follow from the smaller clause 1. Finally, since $g(f(a, b)) \succ g(g(f(b, a)))$, an inference by *Paramodulation left* with 5 into 6 gives us

$$7) \quad f(a, b) \simeq g(g(f(b, a))) \rightarrow$$

but, since $f(a, b) \simeq g(g(f(b, a)))$ does not follow from the positive equations, further inferences on 7 will not lead to the empty clause. Therefore the set S can be saturated without the empty clause being derived. □